2152

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Docket No. **9265.00**

CLAIM FOR BENEFIT OF EARLIER-FILED FOREIGN APPLICATION

Application of

Richard Cudd et al.

Serial No. **09/943,118**

Filed: **August 30, 2001**

NOV 01 2002

Group Art Unit: **2152**

Examiner: **Unknown**

FOR: **DOWNLOADING AND UPLOADING DATA IN INFORMATION NETWORKS**

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on NOV 01 2002 (Date of Deposit).

Shirley Doll

Assistant Commissioner for Patents

Washington, D.C. 20231

Sir:

Applicants wish to claim the benefit of the filing date of the earlier G.B. Application Serial No. **0021372.8**, filed on **September 1, 2000**, recited in the Declaration under the provision of 35 U.S.C. 119, and accordingly, Applicants submit herewith a certified copy of said application.

Respectfully submitted,

Michael Chan

Michael Chan
Reg. No. 33,663
Attorney for Applicant(s)

NCR Corporation, Law Department, WHQ5E
1700 S. Patterson Blvd., Dayton, OH 45479-0001
Tel. No. 937-445-4956/Fax No. 937-445-3733

PATENTS • DESIGNS • COPYRIGHT • TRADE MARKS
The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 07 AUG 2001

Patents Form 1/77

E1 SEP 2000 01SEP00 E564912 P01/7700 0.00-0021372.8

1/77

The Patent Office

Patents Act 1977
(Rule 16)

1-SEP 2001

The Patent Office
Cardiff Road
Newport
South Wales NP9 1RH

Statement of inventorship and of right to grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

1. Your reference

9265

2. Patent application number
(The Patent Office will fill in this part)

0021372.8

3. Full name, address and postcode of the or of each applicant (*underline all surnames*)

NCR INTERNATIONAL, INC
1700 SOUTH PATTERSON BOULEVARD
DAYTON, OHIO 45479
UNITED STATES OF AMERICA

Patents ADP number (*if you know it*)

6105449001

If the applicant is a corporate body, give the country/state of its incorporation

INCORPORATED IN THE STATE OF DELAWARE

4. Title of the invention

DOWNLOADING AND UPLOADING DATA IN INFORMATION NETWORKS

5. Name of your agent (*if you have one*)
"Address for service" in the United Kingdom to which all correspondence should be sent (including the postcode)

F CLEARY
INTERNATIONAL IP DEPARTMENT
NCR LIMITED
206 MARYLEBONE ROAD
LONDON NW1 6LY

Patents ADP number (*if you know it*)

1982003

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (*if you know it*) the or each application number

Country	Priority application number (*if you know it*)	Date of Filing (*day/month/year*)

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing (*day/month/year*)

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (*Answer 'Yes' if:*

a) *any applicant named in part 3 is not an inventor, or*
b) *there is an inventor who is not named as an applicant, or*
c) *any named applicant is a corporate body.*

See note (d))

YES

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document.

Continuation sheets of this form

Description	12
Claim(s)	9
Abstract	1
Drawing(s)	2 x 2

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translation of priority documents	
Statement of inventorship and right to grant of a patent (*Patents Form 7/77*)	
Request for preliminary examination (*Patents Form 9/77*)	1
Request for substantive examination (*Patents Form 10/77*)	
Any other documents (*please specify*)	

11. I/We request the grant of a patent on the basis of this application.

Signature Date 30/08/2000

12. Name and daytime telephone number of person to contact in the United Kingdom

CHRISTINE SHEPPARD
020 7725 8379

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 01645 500505*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

DOWNLOADING AND UPLOADING DATA IN INFORMATION NETWORKS

This invention relates to downloading data from or uploading data to information sources via information networks and, in preferred embodiments, relates to techniques for retrieving files such as web pages and other web content in an Internet environment.

Currently, the Internet operates under the hypertext transfer protocol (HTTP) and embodies a client-server architecture. The vast majority of Internet access - about 99% - is achieved via web browser programs, predominantly Netscape or Microsoft Internet Explorer, whose trade marks are acknowledged.

Existing download techniques will be discussed later with reference to Figures 1(a) and 1(b) but, typically, the client is a user's terminal such as a PC, a suitably-adapted (e.g. Wireless Access Protocol or WAP) mobile telephone or other communications device running a browser program. This terminal downloads and displays a desired HTML web page held on a web server by using a communications network to send a request for that web page across the Internet to the appropriate server. The server responds by sending the requested web page back across the Internet and from there to the client via the communications network to which the user's terminal is connected.

Whilst a web page is mentioned by way of example, other web content files such as .gif, .jpg or .mpg files can be downloaded in the same way.

The client and server can be in direct contact across the Internet via the communications network or can be connected via a proxy server acting between the client and the server. The purpose of the proxy server is to cache some web pages, usually as a result of previous user requests, so that future user requests for the cached web pages can be satisfied without connecting to the server. If the user requests a web page that is not cached on the proxy server, the proxy server forwards the request to the server and receives and forwards the requested page from the server to the client.

However, in general, less traffic needs to connect to the server and so the average download time is decreased.

Cache techniques are, of course, commonplace in the Internet art. Most commonly, when a server or a proxy server responds to a user's request by sending a web page back to the client, that page may be cached on the user's terminal so that future user requests for the same web page can be satisfied immediately without having to connect to the server or the proxy server at all. Nevertheless, the user's terminal cannot cache every page that the user ever downloads, and the user will naturally wish to update cached web pages and to download new web pages from time to time. This means that efficient downloading remains paramount.

Despite ongoing efforts to speed Internet usage with faster modems and high-speed network technologies such as ADSL and optical cable, the majority of Internet users are burdened with slow download times. Even if an Internet user invests heavily in a fast modem and in subscribing to a high-speed communications network, the user may still suffer delays due to the architecture of the Internet itself and the nature of its components. Particular problems arise due to the limited speed with which servers can operate and the restricted bandwidth of the numerous communications channels that lie between the server and the client. There is also the problem of unreliability, meaning that if a server is down and no cached copy of the desired web page is accessible elsewhere, the user may have to wait until the server is operational again.

The slowness and unreliability of downloads makes the Internet less useful and appealing than it could and should be, to the detriment of users and also those who seek to provide information to users. Recent research suggests that, on average, a user will wait just eight seconds for a web page to download before moving on elsewhere. If that happens, the user misses information that could have been beneficial and the provider of the web page misses an opportunity to convey that information, possibly resulting in lost business and decreased advertising revenues. The problem is likely to get worse until efforts to upgrade the Internet and its associated communications technologies begin to outweigh the explosion of new Internet users and the move towards 'always-on' Internet access.

Broadly, this invention contemplates a method of downloading data via a client-server communications network, which network comprises a plurality of clients each having a local cache storing data downloaded via the network. The method comprises responding to a data request made to the network by a first client by uploading data from the cache of a second client and transmitting that data across the network to the first client.

In use of a first architecture, the invention may be defined as a method of downloading or uploading data via a client-server communications network that includes a server and a plurality of clients, each client having a local cache storing data downloaded via the network, the method comprising a requesting client (8) sending a request for data to the server, and the server responding by sending the requested data to the requesting client (8) or referring the requesting client (8) to a proxy server client that holds the requested data in its local cache, the requesting client (8) then downloading the requested data from the cache of the proxy server client across the network.

This first architecture is embodied in a client-server communications network including a server and a plurality of clients, each client having a local cache storing data downloaded via the network and the server having means for responding to a client that sends a data request to the server, wherein the server is adapted to send the requested data to the requesting client (8) or to refer the requesting client (8) to a proxy server client that holds the requested data in its local cache.

The server of the first architecture includes means for responding to a client that sends a data request to the server and is adapted to send the requested data to the requesting client or to refer the requesting client to a proxy server client that holds the requested data in a local cache of data downloaded via the network. The first architecture also involves a client terminal for connection to a server, or a browser for such a client terminal, including selection means for choosing between a plurality of proxy server clients if the server is unable to respond to a data request from the client within a target period or at all, and means for downloading the requested data from a chosen proxy server client.

In use of a second architecture, the invention may be defined as a method of downloading or uploading data via a client-server communications network that includes a server and a plurality of clients, each client having a local cache storing data downloaded via the network, the method comprising a requesting client broadcasting a data request over the network to the server and/or one or more other clients or connecting to at least one client whose address is on a proxy list held by the requesting client, the requesting client then downloading the requested data across the network from the cache of a proxy server client that is caching the requested data.

This second architecture is embodied in a client-server communications network including a server and a plurality of clients, each client having a local cache storing data downloaded via the network, wherein a requesting client is adapted to broadcast a data request over the network to the server and/or one or more other clients or to connect to at least one client whose address is on a proxy list held by the requesting client, and includes means for downloading the requested data across the network from the cache of a proxy server client that is caching the requested data.

The second architecture involves a client terminal for connection to a server, or a browser for such a client terminal, including selection means for choosing among a plurality of proxy server clients the proxy server client from which it will download the requested data, and means for downloading the requested data from a chosen proxy server client.

In an Internet context, the network is the Internet, the clients are user terminals running web browsers and the respective local caches are associated with the browsers on the user terminals that act as proxy server clients. The invention capitalises upon the facts that (i) the vast majority of Internet access is done through web browsers that have become a *de facto* standard, such as Netscape and Microsoft Internet Explorer and (ii) those millions of web browsers cache a great deal of information about the web sites that people most often look at.

A key advantage of the invention, in preferred embodiments, is that it can be deployed and propagated among a large number of users as a plug-in for their existing browsers. So, whilst a simple module would have to be added to the server, users

would not have to install completely new browsers but instead could upgrade their existing systems using simple download techniques with which most users are familiar. The invention therefore extends to software plug-ins for a client terminal or for a browser loaded on that client terminal and being programmed to adapt a terminal or a browser in accordance with the invention.

In order that this invention can be more readily understood, reference will now be made, by way of example, to the accompanying drawings in which:

Figure 1(a) and Figure 1(b) are block diagrams showing existing ways in which a web page may be downloaded via the Internet, Figure 1(a) showing a simple client-server architecture and Figure 1(b) showing a variant in which a proxy server acts between the client and the server;

Figure 2 is a block diagram showing how, in a preferred embodiment of the invention, a web page can be downloaded by one client from another client;

Figure 3 is a block diagram showing how the bandwidths of connections between various clients can be compared;

Figure 4 is a block diagram of a second embodiment of the invention that can be used in isolation or, preferably, as a fall-back for the preferred embodiment.

In the simplest existing client-server architecture of Figure 1(a), a client 1 such as a PC running a browser makes an HTTP request 2 for a particular downloadable file to the server 3 and the server 3 responds by sending the requested file to the client 1. The file could, for example, be an HTML web page 4, a .gif, .jpg or .mpg file, or other web content. Web pages will be used as an example of such files throughout the description that follows.

Figure 1(b) shows a variant in which a proxy server 5 acts between the client 1 and the server 3 to speed the average download time by reducing traffic connecting to the server 3. The client 1 makes an HTTP request for a particular web page via the proxy server 5. The proxy server 5 may have a cached copy of the requested web page as a

result of a previous request for that web page. If it does, then it returns that cached web page to the client 1, without connecting to the server 3. If not, the proxy server 5 requests and downloads the requested web page from the server 3, forwards it to the client 1 and advantageously keeps a copy in its cache in case a client 1 requests that page in the future.

In both of the above variants, the client 1 may cache a downloaded web page so that if a user requests the same page in future, it is not necessary to download that web page again from either a server 3 or a proxy server 5. However, the user may set the browser so that when a web page in cache is requested, the cached web page is compared to the corresponding web page then available from the server 3 or the proxy server 5. If there is a difference between the ostensibly corresponding web pages, the latest version of the web page can be downloaded, displayed and cached in place of the previous version.

Referring now to Figure 2 which illustrates a preferred embodiment of the invention, two clients, Client A and Client B, can access a server 3 via the Internet. There would of course be many more clients in practice, but just two clients are necessary to illustrate the broad inventive concept. Both Client A and Client B run browsers that have been enhanced in accordance with the invention, preferably by downloading and executing a suitable plug-in on the respective client terminals.

In the manner of the prior art, each respective terminal of Client A and Client B holds web pages in cache as a result of previous downloads. However, by virtue of the invention, the enhanced browsers open the caches of their respective client terminals for access by other network users. So, as the client terminals can act like proxy servers, the entire network can offer faster and more reliable downloads. In effect, the invention creates a network in which there are almost as many proxy servers as there are clients. These clients that emulate proxy servers will be referred to herein as proxy server clients.

In the embodiment of Figure 2, each proxy server client reports its cached web pages to the server 3 so that the server 3 can store in a look-up table the client location(s) at which a particular web page is cached. In use, a user at Client A requests a web page

from the server 3 with an HTTP request in the usual way. The server 3 either fulfils that request or, if it cannot fulfil the request quickly enough, looks up where else that web page is cached and responds to Client A with a short acknowledgement that points the browser of Client A to the appropriate proxy server client location. As will be explained, the acknowledgement to the requesting client can be little more than a list of IP addresses constituting a proxy list of clients to identify the proxy server client terminal(s) at which the requested web page is cached.

If the server 3 tells Client A that the web page requested by Client A is cached at Client B, then Client A contacts Client B and downloads that web page from the cache in the client terminal of Client B. Preferably, however, Client A firstly assesses and compares the bandwidths available at that time in the connections between itself and the server 3 on the one hand and between itself and Client B on the other hand. The aim is to determine which of the available connections would be the faster to use, and then to select that connection so as to minimise download times and maximise the efficient use of network resources.

In Figure 2, the bandwidth available between Client A and Client B has been assessed as being greater than the bandwidth available between Client A and the server 3, so Client A requests the desired web page from the cache of Client B. However, if the bandwidth comparison was instead in favour of the connection between Client A and the server 3, that connection would be used to download the web page from the server 3 instead of from Client B.

By extension, the technique of comparing bandwidths can be applied to the connections between Client A and proxy server clients other than Client B. So, if the desired web page is cached at other network resources such as other clients, the web page can be downloaded from one of those other resources if it would be more efficient, bandwidth-wise, to do so than to download the same web page from either Client B or the server 3.

The principle of resource selection is shown in Figure 3, in which Client A has received a proxy list 6 of client IP addresses 7 from the server and then assesses the speed of the connection to each of the proxy server clients identified by the proxy list

6. This is done by a simple PING (Packet INternet Groper) operation that attempts to contact each specified IP address 7 and returns the times taken to connect to the terminals at those addresses 7. Specifically, a PING utility sends a packet to each IP address 7 and waits for, and times, the reply from each address 7. It will be noted that Client A performs the PING operation rather than the server 3 because it is the connections between respective clients that matter in this context, not the connections between the server 3 and its clients.

The response times from each pinged proxy server client are recorded by Client A so that, once an appropriate number of proxy server clients have been pinged, Client A can compare the recorded response times and select the proxy server client with the fastest response for the purpose of downloading the desired web page. It is also possible for a target response time to be set and for Client A to select from the first pinged proxy server client to meet that target. This saves Client A continuing the process of pinging all of the proxy server clients on the proxy list when it has already found a proxy server client whose response time meets the target and so is deemed to be adequate.

It will be apparent that the pinging process set out in Figure 3 is preceded by an assessment of whether it is faster to download the desired web page directly from the server 3 or to download from one of the proxy server clients that are caching the desired web page. This assessment can be done in various ways. In a first technique, for example, the server 3 may initially attempt to respond with the desired web page but if it cannot respond within a predetermined period deemed to be acceptably quick, it instead responds with the aforementioned proxy list and leaves the requesting client to find and download the desired web page from another client. This has the advantage that if the server 3 can respond and upload the desired web page to the requesting client quickly enough, there is no need to go through the delay of pinging other clients at which the desired web page is cached.

The predetermined period in which the server 3 is challenged to respond need not be a fixed period of time: that period could change dynamically in accordance with the average download time for the requesting client terminal concerned. Clearly, all else being equal, a client terminal connected to the Internet via an ordinary 56k modem

will expect slower downloads than a terminal using a modem that can exploit a 128k ISDN or ADSL connection. In those circumstances, it is appropriate that the server 3 senses the speed of the connection from the requesting client and responds to the sensed speed by tailoring the predetermined period accordingly. Specifically, the server 3 should shorten the predetermined period when the connection is relatively fast, and should lengthen that period when the connection is relatively slow.

A second technique for assessing the speed of server response involves the server 3 invariably and immediately responding to web page requests with a proxy list 6 but including in that proxy list 6 the IP address of the server 3 itself. As a result, the server 3 is treated like the other resources (i.e. proxy server clients) identified by the proxy list 6 and so will be pinged along with the proxy server clients identified by that list. If the server 3 happens to be the fastest resource to respond, or if it is the first resource to respond within a predetermined target time, then the desired web page is downloaded from the server 3. Otherwise, the desired web page is downloaded from a proxy server client whose response is found to be fastest among the various resources identified by the proxy list 6 or whose response time is the first to meet the target.

This second assessment technique is currently less preferred than the first assessment technique because although it is more elegant in terms of architecture, it incurs the overhead of downloading the proxy list 6 from the server 3 and then pinging the IP addresses 7 on that list.

The various proxy server client locations, expressed as their respective IP addresses, can be stored on the server 3 as simple text files and a proxy list file is associated with every web page in the look-up table held on the server. The size of the proxy list file should obviously be kept within reasonable bounds, for example limited to a maximum of twenty IP addresses for each web page. This is due to considerations of memory capacity and download time but also has implications for efficient bandwidth assessment, in which it is desirable not to ping too many client terminals.

Another approach of the invention is shown in Figure 4, in which a client 8 making a web page request holds a proxy list of IP addresses defining a server 3 and also a group of other clients 9. The request is made to the IP addresses in the proxy list, so

being made to the server 3 in the normal way but also being broadcast to the group of other proxy server clients 9, to enquire as to whether they hold any of the requested information in cache. If they do, they can report back to the requesting client 8 and the web page can then be downloaded by that client 8 from any of the proxy server clients 9 that are caching the requested web page. Each of the client terminals on the proxy list may in turn be connected to other proxy server client terminals 10 to which they can forward the request, thus forming the chain or tree structure shown in simplified form in Figure 4.

In the Figure 4 approach, the server 3 is not relied upon to return a list of proxy terminals. The requesting client terminal merely needs to have the IP address of one, or the IP addresses of a few, of the proxy server client terminals 9 for the chain or tree to begin. The necessary IP address(es) could be downloaded from a web site or distributed with software.

It would also be possible to broadcast a short request over the network to make an initial connection with a proxy server client terminal 9 that has the requested web page in cache and responds to the broadcast. This operation would only need to be performed once since the IP address of the responding proxy server client terminal could then be stored by the requesting client terminal for later use.

An advantage of the approach of Figure 4 is that it is possible to find web pages even when the server 3 is down. It could thus be a fall-back to the architecture of Figure 2, to be used only when there is no response from the original server in operation of the Figure 2 embodiment.

If appropriate, the choice between potential proxy server client sources 9 of the cached web page can be made after the abovementioned bandwidth comparison between the various connections to those proxy server clients 9. As before, this involves pinging all the proxy server client terminals 9 on the proxy list, including the server 3, and downloading via the fastest connection.

Again, a decision is required about whether to ping each IP address on the proxy list held by the requesting client 8, or simply to download the web page from the

server 3. If the Figure 4 architecture is used as a fall back, then a timeout can be set so that if no response has been received from the server 3 after a predetermined (but not necessarily fixed) period of time, then the chain or tree process is followed.

Another issue with pinging through a chain or tree structure is that each proxy server client terminal 9 at each level of the structure will ping to proxy server client terminals 10 in the next level of the structure. So, the originally-requesting client terminal 8 will not necessarily ping directly to the potential source of the requested web page if that source is more than one level down the structure. In those circumstances, it is necessary to add another step into the process to check the speed between the originally-requesting client terminal 8 and the potential source 10A, as shown in Figure 4, to be sure that there is a fast connection for downloading from the potential source 10A.

In all cases, a client advantageously reports to the server 3 upon downloading and caching a web page retrieved from cache among proxy server clients. In this way, the server can add that proxy server client location to its look-up table as a further potential source of that web page. It is similarly advantageous that a proxy server client reports to the server 3 removal of a web page from its cache, for example during a routine automated cache clean-up or in response to a user command. The server 3 can therefore delete that web page location from its look-up table and so knows to point requesting clients elsewhere if they request the deleted web page.

The invention requires extra messaging but it is expected that, in most practical cases, the overhead of that extra messaging in terms of download time would still allow shorter aggregate download times than can be achieved directly from the server.

The inventors recognise the need to ensure that no proxy server clients are overloaded with incoming requests, broadcasts and the resulting uploads, as this would unacceptably reduce the bandwidth available to those clients for other, unrelated communications. Accordingly, the invention contemplates means for monitoring the proxy workload of proxy server clients and preventing overload. This can be achieved at the client end by refusing to serve certain requests over a specified workload limit,

and/or at the server end by omitting an overloaded proxy server client from the proxy lists sent in response to requests from other clients. Also, in the embodiment of Figure 4, the invention contemplates limiting the number of proxy server clients in the group of clients that are polled and, possibly, changing the members of that group from request to request.

Many other variations are possible within the inventive concept. For example, the server in the Figure 2 embodiment can periodically update the look-up table from which the proxy lists are derived so to ensure that the lists are optimal. This can be achieved by pinging the IP addresses in the table from time to time, comparing their response times and discarding the slowest for a given item of data or those that fail to meet a target threshold. A possible problem with this approach is that the connection between a server and a proxy server client does not equate to the connection between one client and another client, so the server is not always best placed to assess client-client bandwidth. Another and possibly better approach is that when a client reports to the server upon downloading and caching a web page received from a resource such as a proxy server client, that client tells the server the IP address of the proxy server client that provided the web page. The server can therefore assemble a list of the most commonly accessed and hence fastest resources and can discard the less commonly accessed and hence slowest resources, like a voting scheme in which only those resources proven by various requesting clients to be fastest will continue to survive in the proxy lists held by the server.

The present invention may be embodied in other specific forms without departing from its essential attributes. Accordingly, reference should be made to the appended claims rather than to the foregoing specific description as indicating the scope of the invention.

CLAIMS

1. A method of downloading or uploading data via a client-server communications network that includes a server (3) and a plurality of clients (A, B), each client (A, B) having a local cache storing data downloaded via the network, the method comprising a requesting client (A) sending a request for data to the server (3), and the server (3) responding by sending the requested data to the requesting client (A) or referring the requesting client to a proxy server client (B) that holds the requested data in its local cache, the requesting client (A) then downloading the requested data from the cache of the proxy server client (B) across the network.

2. The method of Claim 1, wherein the network is the Internet, the clients (A, B) are user terminals running web browsers and the respective local caches are associated with the browsers on the user terminals.

3. The method of Claim 1 or Claim 2, wherein the server (3) refers the requesting client (A) to the proxy server client (B) if the server (3) cannot send the requested data to the requesting client (A) within a predetermined target period.

4. The method of Claim 3, wherein the predetermined target period is variable.

5. The method of Claim 4, wherein the predetermined target period is varied in accordance with the speed of the connection to the requesting client (A).

6. The method of any preceding Claim, wherein the server (3) refers the requesting client (A) to a plurality of proxy server clients (B) and the requesting client (A) chooses from that plurality the proxy server client (B) from which it will download the requested data if that data is not provided by the server (3).

7. The method of Claim 6, wherein the server (3) sends the requesting client (A) a proxy list (6) containing the respective addresses of the plurality of proxy server clients (B).

8. The method of Claim 7, wherein the proxy list (6) also contains the address of the server (3).

9. The method of Claim 7 or Claim 8, wherein the requesting client (A) assesses the speed of at least one connection to an address in the proxy list (6).

10. The method of Claim 9, wherein the requesting client (A) assesses the speed of the connection by pinging the associated address taken from the proxy list (6).

11. The method of Claim 9 or Claim 10, wherein the requesting client (A) assesses the speeds of connections to more than one address in the proxy list (6), records and compares the measured speeds, and downloads the requested data from the address with the fastest connection.

12. The method of any of Claims 9 to 11, wherein the requesting client (A) sets a target connection speed and downloads the requested data from the first address in the proxy list (6) to meet that target.

13. The method of any preceding Claim, wherein the server (3) maintains a look-up table correlating items of data with the addresses of proxy server clients (B) that are caching that data.

14. The method of Claim 13, wherein proxy server clients (B) report changes in their cache status to the server (3).

15. The method of Claim 14, wherein the requesting client (A) reports to the server (3) that it has downloaded and is caching the requested data and the server (3) associates the address of the requesting client (A) with the requested data in the look-up table so as to use the requesting client (A) as a proxy server client (B) for future requests by other clients for the requested data.

16. The method of any of Claims 13 to 15, wherein the server (3) updates the look-up table by assessing connection speeds to the proxy server client addresses in the table

and discarding or demoting the addresses of proxy server clients (B) to which connections are relatively slow.

17. The method of Claim 16, wherein the addresses in the table are pinged and their response times are compared with each other or with a target threshold.

18. The method of Claim 16, wherein the requesting client (A) reports to the server (3) the address of the proxy server client (B) that provided the requested data, and wherein the server (3) assembles an address list of the proxy server clients (B) most commonly accessed to obtain the requested data.

19. The method of any preceding Claim, wherein if the server (3) fails to respond within a predetermined target period, the requesting client (A) broadcasts the data request over the network to a plurality of other clients or connects to at least one proxy server client (B) whose address is on an auxiliary proxy list held by the requesting client (A).

20. The method of Claim 19, wherein the auxiliary proxy list contains the respective addresses of a plurality of proxy server clients (B).

21. The method of Claim 19 or Claim 20, wherein the requesting client (A) assesses the speed of at least one connection to an address in the auxiliary proxy list.

22. The method of Claim 21, wherein the requesting client (A) assesses the speed of the connection by pinging the associated address taken from the auxiliary proxy list.

23. The method of Claim 21 or Claim 22, wherein the requesting client (A) assesses the speeds of connections to more than one address in the auxiliary proxy list, records and compares the measured speeds, and downloads the requested data from the address with the fastest connection.

24. The method of any of Claims 20 to 23, wherein the requesting client (A) sets a target connection speed and downloads the requested data from the first address in the auxiliary proxy list to meet that target.

25. The method of Claim 19 or Claim 20, wherein the data request cascades through levels of a tree or chain client structure to a proxy server client (B) whose address is not on the auxiliary proxy list held by the requesting client (A).

26. The method of Claim 25, wherein the requesting client (A) checks that the speed of the connection to the proxy server client (B) meets a target connection speed before downloading the requested data from the proxy server client (B).

27. The method of Claim 26, wherein the requesting client (A) assesses the speed of the connection by pinging the proxy server client (B) and comparing the measured connection speed with the target connection speed.

28. The method of any preceding Claim, wherein proxy server clients (B) maintain a workload limit and do not serve requests that would exceed that workload limit.

29. The method of any preceding Claim, wherein the server (3) monitors the workload of proxy server clients (B) and does not refer the requesting client (A) to proxy server clients (B) whose workload is above a workload limit.

30. A client-server communications network including a server (3) and a plurality of clients, each client having a local cache storing data downloaded via the network and the server (3) having means for responding to a client that sends a data request to the server (3), wherein the server (3) is adapted to send the requested data to the requesting client (A) or to refer the requesting client (A) to a proxy server client (B) that holds the requested data in its local cache.

31. The network of Claim 30, wherein the network is the Internet, the clients are user terminals running web browsers and the respective local caches are associated with the browsers on the user terminals.

32. The network of Claim 30 or Claim 31, wherein the server (3) is responsive to timeout means to refer the requesting client (A) to the proxy server client (B) if the

server (3) cannot send the requested data to the requesting client (A) within a predetermined target period.

33. The network of Claim 32, wherein the timeout means includes means for varying the predetermined target period.

34. The network of Claim 33, wherein the timeout means includes means for sensing the speed of the connection to the requesting client (A) and for varying the predetermined target period in accordance with the sensed connection speed.

35. The network of any of Claims 30 to 34, wherein the requesting client (A) includes selection means for choosing between a plurality of proxy server clients (B) referred by the server (3).

36. The network of Claim 35, wherein the selection means selects from a proxy list (6) containing the respective addresses of the plurality of proxy server clients (B) and optionally also containing the address of the server (3).

37. The network of Claim 36, wherein the selection means comprises bandwidth assessment means for measuring the speed of connection to an address in the proxy list (6).

38. The network of Claim 37, wherein the bandwidth assessment means includes means for pinging an address taken from the proxy list (6).

39. The network of Claim 37 or Claim 38, wherein the bandwidth assessment means includes means for recording speeds of connections to addresses on the proxy list (6), means for comparing the recorded speeds, and means for downloading the requested data from the address with the fastest recorded speed.

40. The network of any of Claims 37 to 39, wherein the selection means includes means for comparing a measured connection speed to an address with a target connection speed and downloading the requested data from that address if the measured connection speed meets the target connection speed.

41. The network of any of Claims 30 to 40, wherein the server (3) stores a look-up table correlating items of data with the addresses of proxy server clients (B) that are caching that data.

42. The network of Claim 41, wherein proxy server clients (B) include means for reporting changes in their cache status to the server (3).

43. The network of Claim 41 or Claim 42, wherein the server (3) includes means for updating the look-up table, the updating means including means for assessing connection speeds to the proxy server client addresses in the table and means for discarding or demoting the addresses of proxy server clients (B) to which connections are relatively slow.

44. The network of Claim 43, wherein the server (3) includes means for pinging the addresses in the table, means for recording their response times and means for comparing the recorded response times with each other or with a target threshold.

45. The network of Claim 43 or Claim 44, wherein the requesting client (A) includes means for reporting to the server (3) the address of a proxy server client (B) that provides the requested data, and wherein the server (3) includes means for assembling an address list of the proxy server clients (B) most commonly accessed to obtain the requested data.

46. The network of any of Claims 30 to 45, wherein the requesting client (A) includes timeout means responsive to failure by the server (3) to respond within a predetermined target period, and means responsive to the timeout means to broadcast a data request over the network to a plurality of other clients or to connect to at least one proxy server client (B) whose address is on an auxiliary proxy list held by the requesting client (A).

47. The network of Claim 46, wherein the proxy server clients (B) are in a tree or chain structure comprising a plurality of levels.

48. The network of any of Claims 30 to 47, wherein a proxy server client (B) includes workload limit means that prevents the proxy server client (B) serving requests that would exceed a workload limit.

49. The network of any of Claims 30 to 48, wherein the server (3) includes workload limit means that prevents referral of the requesting client (A) to proxy server clients (B) whose workload is above a workload limit.

50. A server (3) for a client-server communications network, the server (3) including means for responding to a client that sends a data request to the server (3) and being adapted to send the requested data to the requesting client (A) or to refer the requesting client (A) to a proxy server client (B) that holds the requested data in a local cache of data downloaded via the network.

51. The server (3) of Claim 50 and being responsive to timeout means to refer the requesting client (A) to the proxy server client (B) if the server (3) cannot send the requested data to the requesting client (A) within a predetermined target period.

52. The server (3) of Claim 51, wherein the timeout means includes means for varying the predetermined target period.

53. The server (3) of Claim 52, wherein the timeout means includes means for sensing the speed of the connection to the requesting client (A) and for varying the predetermined target period in accordance with the sensed connection speed.

54. The server (3) of any of Claims 50 to 53, including a stored look-up table correlating items of data with the addresses of proxy server clients (B) that are caching that data.

55. The server (3) of Claim 54 and including means for updating the look-up table, wherein the updating means comprises means for assessing connection speeds to the proxy server client addresses in the table and means for discarding or demoting the addresses of proxy server clients (B) to which connections are relatively slow.

56. The server (3) of Claim 55, including means for pinging the addresses in the table, means for recording their response times and means for comparing the recorded response times with each other or with a target threshold.

57. The server (3) of any of Claims 50 to 56, including means for assembling an address list of proxy server clients (B) most commonly accessed to obtain the requested data.

58. The server (3) of any of Claims 50 to 57, including workload limit means that prevents referral of the requesting client (A) to proxy server clients (B) whose workload is above a workload limit.

59. A client terminal for connection to a server (3), or a browser for such a client terminal, including selection means for choosing between a plurality of proxy server clients (B) if the server (3) is unable to respond to a data request from the client within a target period or at all, and means for downloading the requested data from a chosen proxy server client (B).

60. The terminal or browser of Claim 59, wherein the selection means selects from a proxy list (6) containing the respective addresses of the plurality of proxy server clients (B) and optionally also containing the address of the server (3).

61. The terminal or browser of Claim 59 or Claim 60, wherein the selection means comprises bandwidth assessment means for measuring the speed of connection to an address in the proxy list (6).

62. The terminal or browser of Claim 61, wherein the bandwidth assessment means includes means for pinging an address taken from the proxy list (6).

63. The terminal or browser of Claim 61 or Claim 62, wherein the bandwidth assessment means includes means for recording speeds of connections to addresses on the proxy list (6), means for comparing the recorded speeds, and means for downloading the requested data from the address with the fastest recorded speed.

64. The terminal or browser of any of Claims 60 to 63, wherein the selection means includes means for comparing a measured connection speed to an address with a target connection speed and downloading the requested data from that address if the measured connection speed meets the target connection speed.

65. The terminal or browser of any of Claims 60 to 64, including means for reporting changes in its cache status to the server (3).

66. The terminal or browser of any of Claims 60 to 65, including means for reporting to the server (3) the address of a proxy server client (B) that provides requested data.

67. The terminal or browser of any of Claims 60 to 66, including timeout means responsive to failure by the server (3) to respond within a predetermined target period, and means responsive to the timeout means to broadcast a data request over the network to a plurality of other clients or to connect to at least one proxy server client (B) whose address is on an auxiliary proxy list held by the requesting client (A).

68. The terminal or browser of any of Claims 60 to 67, including workload limit means that prevents the terminal serving requests that would exceed a workload limit.

69. A software plug-in for a client terminal or for a browser loaded on that client terminal and being programmed to adapt the terminal or the browser in accordance with any of Claims 59 to 68.

DOWNLOADING AND UPLOADING DATA IN INFORMATION NETWORKS

Abstract

A method of downloading or uploading data via a client-server communications network, which network comprises a plurality of clients (A, B, 8, 9, 10) each having a local cache storing data downloaded via the network. The method comprises responding to a data request made to the network by a first client (A, 8) by uploading data from the cache of a second client (B, 9, 10) and transmitting that data across the network to the first client (A, 8). Also disclosed are client-server networks operating in accordance with the method and to the related servers (3), client terminals (A, B, 8, 9, 10), browsers loaded on client terminals, and plug-ins for such terminals and browsers.

PRIOR ART

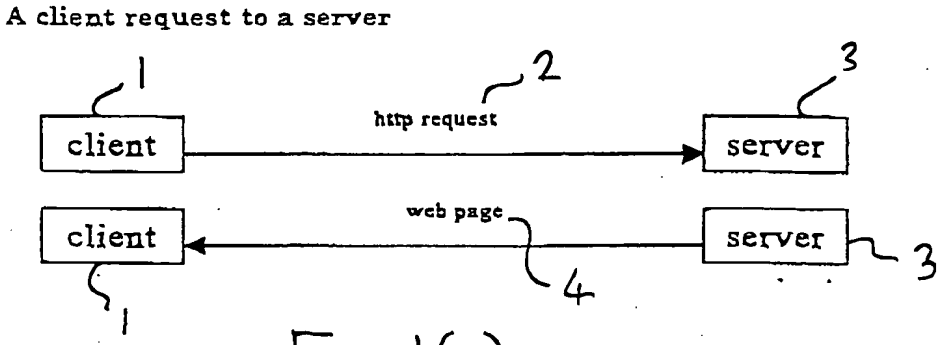

FIG. 1(a)

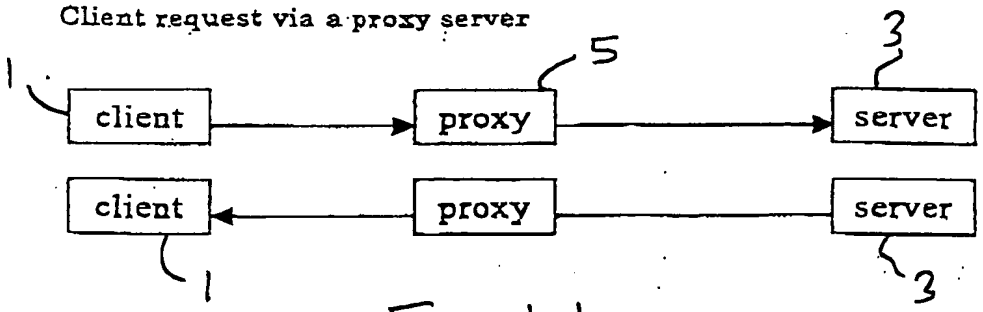

FIG. 1(b)

2/2

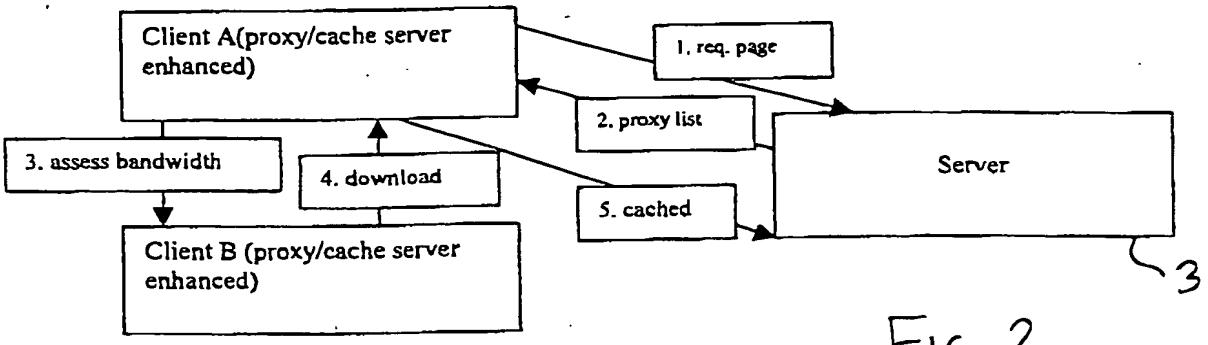

FIG. 2

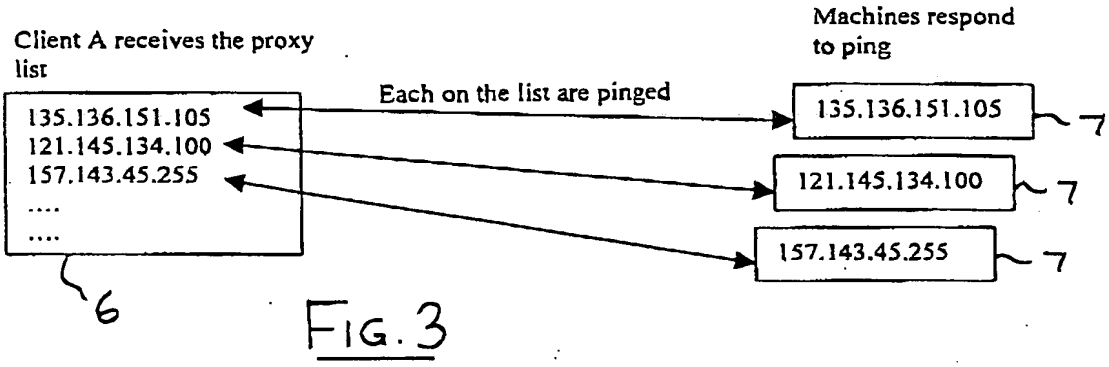

FIG. 3

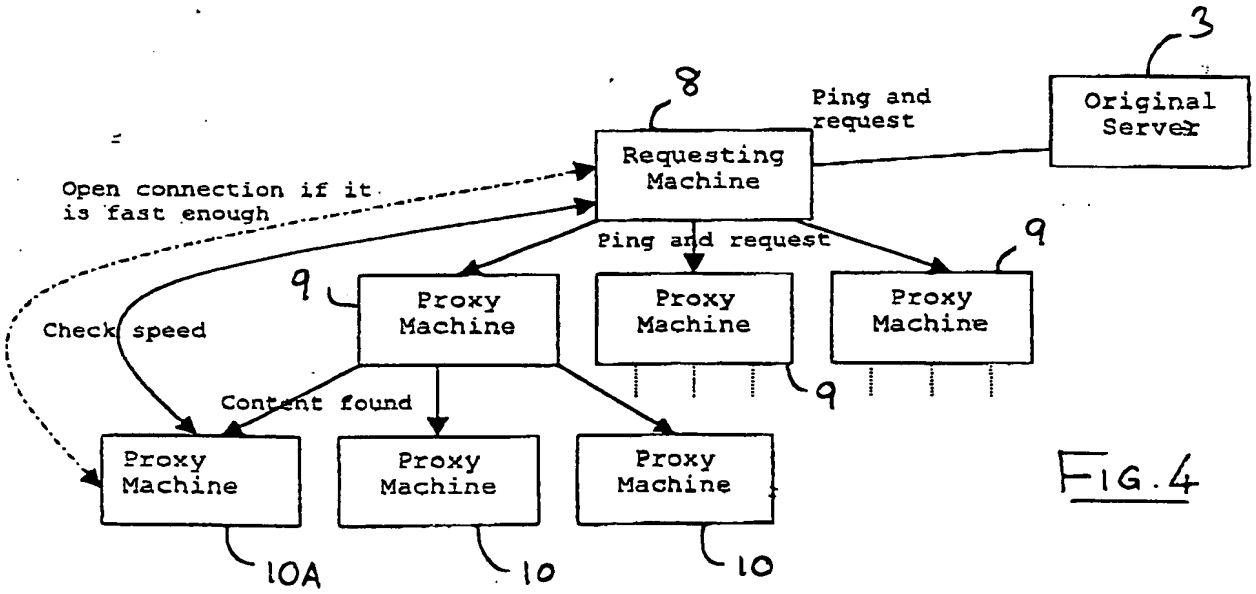

FIG. 4